CLAIMS

1. A computer-implemented information reservoir creation process wherein:

a table collection is constructed from a data source;

said table collection includes a subset of tables designated as sampling initiation tables;

each table in said table collection is a member of either a directly-sampled table set or a descendent-sampled table set;

said directly-sampled table set is characterized by tables that are either sampling initiation tables or ancestor tables to one or more sampling initiation tables;

said descendant-sampled table set is characterized by tables that are descendant tables to a sampling initiation table;

said table collection is characterized by a table collection schema equivalent to a data source schema of said data source, with the exception that a list of attributes for each table of said directly-sampled table set includes an additional attribute containing actual rate of inclusion values;

each tuple included in said table collection is equivalent to one and only one tuple in the corresponding table of said data source;

an actual rate of inclusion value stored with a select data source tuple and included in a directly-sampled table of said table collection represents the probability that a randomly selected table collection produced by the process will contain said select data source tuple.

2. A computer-implemented information reservoir creation process as claimed in claim 1 wherein each tuple included in said table collection is equivalent to one and only one tuple in the corresponding table of said data source.

3. A computer-implemented information reservoir creation process as claimed in claim 1 wherein each tuple included in said table collection is equivalent to one and only one tuple in the corresponding table of said data source after elimination of said actual rate of inclusion value.

4. A computer-implemented information reservoir creation process as claimed in claim 1 wherein said table collection includes all ancestor tuples of each tuple included in any directly-sampled table of the table collection.

5. A computer-implemented information reservoir creation process as claimed in claim 1 wherein said table collection includes all descendant tuples of each tuple included in any sampling initiation table of the table collection.

6. A computer-implemented information reservoir creation process as claimed in claim 1 wherein said probability that a randomly selected table collection produced by the process will contain a given data source tuple in a descendant-sampled table is equal to the actual rate of inclusion stored with a corresponding single ancestor tuple residing in a sampling initiation table.

7. A computer-implemented information reservoir creation process as claimed in claim 1 wherein no pair of data source tuples within any select tuple set taken from directly-sampled tables has an ancestor-descendant relationship.

8. A computer-implemented information reservoir creation process as claimed in claim 7 wherein the probability that a randomly selected table collection produced by the process will contain all of the tuples in said select tuple set is equal to the product of the corresponding actual rates of inclusion associated with each of the individual data source tuples.

9. A computer-implemented method for constructing a representation from a data source in order to provide relatively quick response to queries related to information in said data source, wherein said data source has a plurality of tuples stored in said data source and a data source schema that includes defined relationships among at least a subset of the tuples in the data source, said method comprising:

creating said representation by copying at least a subset of said data source schema to define a representation schema;

adding additional data to said representation that represents information that is not in said data source;

defining tuples of interest within said data source and a degree of interest for each tuple of interest;

sampling tuples from said tuples of interest into said representation based upon said degree of interest in a manner that preserves at least a subset of said relationships among tuples in the data source; and

storing values in the representation that relate to a likelihood that each tuple sampled into said representation would be sampled into the representation if the sampling process were to be repeated.

10. A computer-implemented method as claimed in claim 9 wherein said data source is a table collection.

11. A computer-implemented method as claimed in claim 10 wherein said table collection is a relational database and said defined relationships among tuples are foreign key relationships.

12. A computer-implemented method as claimed in claim 9 wherein said representation schema comprises a logically limited subset of said data source schema.

13. A computer-implemented method as claimed in claim 9 wherein said additional data for an individual tuple includes selected aggregates of descendant tuples.

14. A computer-implemented method as claimed in claim 9 wherein:

said representation is to be used to respond to queries against a parent table that are restricted to parents of a particular kind of child type; and

said representation further includes data added to said representation that is indicative of whether a select tuple in said parent table is associated with said particular kind of child type.

15. A computer-implemented method as claimed in claim 9 wherein said tuples of interest are defined by a plurality of attributes and only a subset of said plurality of attributes are copied for each tuple into said representation.

16. A computer-implemented method as claimed in claim 9 wherein said tuples of interest are defined by associating with each tuple of interest a target rate of inclusion greater than zero and said degree of interest is indicated by the magnitude of the target rate of inclusion.

17. A computer-implemented method as claimed in claim 16 wherein determining said target rate of inclusion comprises taking a minimum of the quantity one and the result of dividing the number of tuples desired in the representation by the total number of tuples in the data source that are to be considered for sampling.

18. A computer-implemented method as claimed in claim 16 wherein said representation is biased by assigning a higher target rate of inclusion for a subset of said tuples of interest.

19. A computer-implemented method as claimed in claim 16 wherein determining said target rate of inclusion comprises taking the minimum of the quantity one and the result of dividing the number of tuples desired in the representation by a number of subpopulations, and dividing that result by the number of tuples in each subpopulation.

20. A computer-implemented method as claimed in claim 16 further comprising:

identifying one or more real-valued attributes of interest in said data source;

clustering said data source based upon said real-valued attributes of interest; and

partitioning said population into subpopulations based upon said clustering, wherein said rates of inclusion are assigned to tuples by subpopulation.

21. A computer-implemented method as claimed in claim 16 wherein said target rate of inclusion is set to its maximum value for tuples containing attribute values that have a high degree of influence on anticipated query results.

22. A computer-implemented method as claimed in claim 16 wherein knowledge of an anticipated workload is encoded into a first set of queries that are representative of said knowledge of said anticipated workload to derive weighting factors used to establish said target rates of inclusion.

23. A computer-implemented method as claimed in claim 22 further comprising:

determining a training set of queries defining a reservoir training set;

associating a set of aggregates with each training query;

collecting said aggregates into a superset;

determining weights for said aggregates in said superset to reflect the importance to users of said representation;

determining a tuning parameter from said weights;

partitioning a sampling population into at least those tuples in the scope of said aggregates, and those tuples outside the scope of said aggregates; and

determining target rates of inclusion for the tuples in each group.

24. A computer-implemented method as claimed in claim 23 wherein said target rates of inclusion for said tuples in the scope of said aggregates in said superset are chosen to minimize the variances of aggregate estimates computed from the representation.

25. A computer-implemented method as claimed in claim 23 wherein said rate of inclusion for tuples participating in sums has the property that tuples with attribute values that are relatively large in magnitude are assigned larger target rates of inclusion.

26. A computer-implemented method as claimed in claim 23 wherein said rate of inclusion for tuples participating in averages has the property that tuples with outlying attribute values are assigned larger target rates of inclusion.

27. A computer-implemented method as claimed in claim 16 further comprising controlling the size of said representation by:

establishing a target number of tuples for said representation;

assigning a tuple preference factor to each tuple among said tuples of interest; and

computing said target rate of inclusion for a select tuple among said tuples of interest based upon said target number of tuples and said tuple preference factor.

28. A computer-implemented method as claimed in claim 27 wherein said tuple preference factor is selected between the values of zero and the quotient defined by the number of said tuples of interest in said data source divided by said target number of tuples such that the sum of all tuple preference factors equals the number of said tuples of interest.

29. A computer-implemented method as claimed in claim 27 wherein said target rate of inclusion for a select tuple among said tuples of interest is computed by multiplying said target number of tuples by said tuple preference factor, and dividing that product by the number of said tuple of interest.

30. A computer-implemented method as claimed in claim 9 wherein the space required by said representation is determined comprising:

determining an average tuple inclusion probability; and

approximating said space by multiplying said average tuple inclusion probability by the sum of a first space required to store the actual tuples in said data source to be considered for sampling and a second space required to store auxiliary structures whose sizes are proportional to said first space, and adding to that product, a third

space required to store auxiliary structures whose sizes are not proportional to said first space.

31. A computer-implemented method as claimed in claim 30 wherein said average tuple inclusion probability is determined by dividing a target number of tuples in said representation by the number of said tuples of interest in said data source.

32. A computer-implemented method as claimed in claim 9 further comprising determining an estimate of the size of said representation by:

obtaining the number of child tuples for a single relationship;

determining whether a target or an induced inclusion probability dominates;

calculating an average actual inclusion probability of a parent table; and

repeating the above steps recursively until an estimate of the expected size of said representation results.

33. A computer-implemented method as claimed in claim 32 wherein the number of child tuples is obtained using a frequency table.

34. A computer-implemented method as claimed in claim 32 wherein the number of child tuples is obtained using an index on the foreign key linking said relationship to said child tuples.

35. A computer-implemented method as claimed in claim 32 wherein said average actual inclusion probability of said parent table is calculated as a weighted average of the average inclusion probability of each subset of parent tuples having the same number of child tuples.

36. A computer-implemented method as claimed in claim 9 wherein ancestor tuples, both within and outside of said tuples of interest, of at least a subset of tuples selected into said representation may be given a higher chance of being selected into said representation.

37. A computer-implemented method as claimed in claim 36 wherein ancestor tuples of at least a subset of tuples selected into said representation are necessarily included in said representation.

38. A computer-implemented method as claimed in claim 9 wherein descendant tuples, both within said tuples of interest and outside of said tuples of interest, of at least a subset of tuples selected into said representation are given a higher chance of being selected into said representation.

39. A computer-implemented method as claimed in claim 38 wherein descendant tuples of at least a subset of tuples selected into said representation are included in said representation

40. A computer-implemented method as claimed in claim 9 wherein an adjusted rate of inclusion is determined for each tuple of interest, said adjusted rate comprising possible contributions from said degree of interest in said tuple, from the results of sampling ancestor tuples of said tuple, and from the results of sampling descendant tuples of said tuple, and the act of sampling an individual tuple among said tuples of interest comprises:

considering a select tuple from said tuples of interest;

simulating a trial in which an event occurs with probability equal to the adjusted rate of inclusion;

determining whether or not the event has occurred; and

copying select tuple into said representation if and only if said event occurs.

41. A computer-implemented method as claimed in claim 40 wherein said event is that a uniform random number on the open interval (0,1) is less than said adjusted rate of inclusion.

42. A computer-implemented method as claimed in claim 40 wherein said trials for any pair of tuples within a table are simulated independently.

43. A computer-implemented method as claimed in claim 40 wherein said act of determining an adjusted rate of inclusion comprises:

assigning a target rate of inclusion to the select tuple of interest;

computing an induced rate of inclusion that represents the rate of inclusion induced by any descendant or ancestor tuples of said select tuple, said induced rate of inclusion set to zero if said select tuple has no descendants or ancestors; and

computing an adjusted rate of inclusion based upon said target rate of inclusion and said induced rate of inclusion, wherein said tuples of interest are sampled based upon said adjusted rate of inclusion.

44. A computer-implemented method as claimed in claim 43 wherein said induced rate of inclusion and said adjusted rate of inclusion are computed only if said select tuple is related to any descendant or ancestor tuples.

45. A computer-implemented method as claimed in claim 43 wherein said tuple of interest is associated with descendant and ancestor tuples that are partitioned into subgroups and said induced rate of inclusion is determined by:

computing an induced rate of inclusion for each subgroup based on the actual rates of inclusion associated with descendant and ancestor tuples in the subgroup; and

computing an overall induced rate of inclusion from the component rates of inclusion induced by each subgroup.

46. A computer-implemented method as claimed in claim 45 wherein said data source is dynamic with new tuples arriving over time, wherein each subgroup comprises sibling tuples partitioned by their arrival time into said data source.

47. A computer-implemented method as claimed in claim 45 wherein said data source is distributed over a number of computer devices greater than one, wherein each subgroup comprises sibling tuples partitioned by computer devices.

48. A computer-implemented method as claimed in claim 43 wherein said adjusted rate of inclusion is equal to the greater of zero and the result of the induced rate of inclusion subtracted from the target rate of inclusion divided by the result of subtracting the induced rate of inclusion from one.

49. A computer-implemented method as claimed in claim 43 wherein said select tuple is sampled at the time said select tuple's corresponding table is sampled at a sampling rate equal to the adjusted rate of inclusion.

50. A computer-implemented method as claimed in claim 43 wherein said select tuple is not sampled if said induced rate of inclusion is greater than or equal to said target rate of inclusion.

51. A computer-implemented method as claimed in claim 9 wherein an actual rate of inclusion is computed for each tuple selected into said representation, said actual rate of inclusion reflecting all opportunities for said tuple to be included in said representation.

52. A computer-implemented method as claimed in claim 51 wherein said actual rate of inclusion is part of said additional data added to said representation.

53. A computer-implemented method as claimed in claim 9 wherein said method further comprises:

representing said subset of said data source schema as a directed, acyclic graph having tables as vertices and table relationships as directed edges, said edges defining ancestor-descendant relationships between tuples in said data source;

traversing said vertices of said acyclic graph;

sampling each tuple associated with said vertices as each vertex is visited;

copying each tuple selected through sampling into said representation; and

optionally copying ancestor and descendant tuples associated with each tuple selected through sampling into said representation.

54. A computer-implemented method as claimed in claim 53 wherein said data source is a table collection.

55. A computer-implemented method as claimed in claim 54 wherein said table collection is a relational database and said ancestor-descendant relationships between tuples are foreign key relationships.

56. A computer-implemented method as claimed in claim 53 wherein said act of traversing said vertices comprises:

identifying a subset of the vertices as sampling initiation points;

performing a breadth-first traversal of those vertices identified as sampling initiation points;

traversing all vertices that can be reached from a sampling initiation point via pathways that follow the direction of said directed edges; and

traversing all vertices that can be reached from a sampling initiation point via pathways that follow the opposite direction of said directed edges.

57. A computer-implemented method as claimed in claim 9 wherein said representation defines a second representation that is a subsample of a first representation, and said method further comprises:

constructing said first representation;

defining subsample tuples of interest within said first representation and a subsample target rate of inclusion for each tuple of interest within said first representation;

constructing said second representation by sampling said first representation according to said subsample target rates of inclusion;

determining a subsample actual rate of inclusion for each tuple included in said second representation; and

determining the actual rate of inclusion for a select tuple in said second representation based on the actual rate of inclusion of said select tuple in said first representation and the subsample actual rate of inclusion of said select tuple in said second representation.

58. A computer-implemented method as claimed in claim 9 wherein said representation defines a third representation that is the union of a first representation and a second representation, and said method further comprises:

constructing said first representation;

constructing said second representation as a result of a sampling process that is independent of the sampling process for said first representation;

constructing said third representation by including any tuple that is included in either said first representation or said second representation; and

determining the actual rate of inclusion for a select tuple in said third representation based on the actual rate of inclusion of said select tuple in said first representation and the actual rate of inclusion of said select tuple in said second representation.

59. A computer-implemented method as claimed in claim 9 wherein said representation defines a third representation that is the intersection of a first representation and a second representation, and said method further comprises:

constructing said first representation;

constructing said second representation as a result of a sampling process that is independent of the sampling process for said first representation;

constructing said third representation by including any tuple that is included in both said first representation and said second representation; and

determining the actual rate of inclusion for a select tuple in said third representation based on the actual rate of inclusion of said select tuple in said first representation and the actual rate of inclusion of said select tuple in said second representation.

60. A computer-implemented method as claimed in claim 9 wherein said representation defines a first representation and said method further comprises establishing a maximum size for said representation and when said maximum size is exceeded, reducing the size of said representation by:

assigning a subsampling target rate of inclusion to each tuple in said first representation;

constructing a second representation by sampling said first representation according to said subsample target rates of inclusion;

determining a subsample actual rate of inclusion for each tuple included in said second representation;

determining the actual rate of inclusion for a select tuple in said second representation based on the actual rate of inclusion of said select tuple in said first representation and the subsample actual rate of inclusion of said select tuple in said second representation; and

replacing said first representation by said second representation.

61. A computer-implemented method as claimed in claim 60 wherein said subsample target rate of inclusion is equal to the desired size of said second representation divide by the size of said first representation.

62. A computer-implemented method as claimed in claim 61 wherein said size is measured in units of numbers of tuples.

63. A computer-implemented method as claimed in claim 61 wherein said size is measured in terms of bytes of disk storage space.

64. A computer-implemented method as claimed in claim 9 further comprising updating said representation in view of a change occurring to said data source, wherein said act of updating comprises:

identifying said change in said data source;

identifying a corresponding tuple in said representation that is associated with said change;

modifying said corresponding tuple in said representation if said change in said data source is a modification and said corresponding tuple exists in said representation; and

deleting said corresponding tuple in said representation if said change in said data source is a deletion and said corresponding tuple exists in said representation.

65. A computer-implemented method as claimed in claim 64 wherein changes are identified based upon a batch driven process.

66. A computer-implemented method as claimed in claim 64 wherein changes are identified in at least near real time.

67. A computer implemented method as claimed in claim 9 further comprising updating said representation in view of added tuples occurring to said data source, wherein said act of updating said representation in view of added tuples comprises:

assigning a rate of inclusion to select ones of said tuples added to the data source; and

sampling from said select ones of said tuples added into said representation based upon associated rates of inclusion.

68. A computer-implemented method as claimed in claim 67 further comprising adjusting select inclusion probabilities over time in response to modifications to said data source.

69. A computer-implemented method as claimed in claim 67 wherein said act of sampling from said added tuples comprises:

constructing a buffer that substantially mirrors said representation schema;

copying said added tuples into said buffer;

copying any ancestor tuples and descendant tuples related to each added tuple into said buffer;

assigning a rate of inclusion to said added tuples in said buffer; and

sampling tuples from said buffer into said representation based upon associated rates of inclusion.

70. A computer-implemented method as claimed in claim 9 further comprising maintaining the relative size of said representation by:

identifying bounds for said representation;

identifying a change to said data source;

updating said representation based upon said change to said data source;

performing a first set of operations if said representation is below said bounds comprising drawing a supplementary sample from said data source and joining said supplementary sample to said representation if deletions to said data source occur more frequently than additions to said data source;

performing a second set of operations if said representation is within said bounds comprising allowing maintenance to said representation based upon said update; and

performing a third set of operations if said representation is above said bounds comprising assigning a deletion inclusion probability to each tuple in said representation and subsampling said representation based upon said deletion inclusion probabilities.

71. A computer-implemented method as claimed in claim 9 wherein said representation is incrementally updated as said data source is updated.

72. A computer-implemented method as claimed in claim 9 wherein said representation is continually rebuilt.

73. A computer-implemented method as claimed in claim 72 wherein said representation is continually rebuilt by defining logical partitions of tables of said data source, ordering said logical partitions, and, for each logical partition:

loading a select partition into a buffer;

adding tuples to said buffer as necessary for said buffer to contain the closure of said select partition;

sampling said buffer;

joining the sampled buffer with said representation; and

updating rates of inclusion of tuples sampled from said buffer.

74. A computer-implemented method as claimed in claim 72 wherein said representation is subsampled to control the size of the rebuilt representation.

75. A computer-implemented method as claimed in claim 9 further comprising answering queries against said data source with approximate answers computed from said representation.

76. A computer-implemented method as claimed in claim 75 further comprising providing a variance with said approximate answer.

77. A computer-implemented method as claimed in claim 75 further comprising providing a confidence interval for the exact answer with said approximate answer.

78. A system for constructing a representation from a data source in order to provide response to queries related to information in said data source, wherein said data source has a plurality of tuples stored in said data source and a data source schema that includes defined relationships among at least a subset of the tuples in the data source, said system comprising:

at least one processor;

at least one storage device communicably coupled to said at least one processor arranged to store said data source and said representation; and

software executable by said at least one processor for:

creating said representation by copying at least a subset of said data source schema to define a representation schema;

adding additional data to said representation that represents information that is not in said data source;

defining tuples of interest within said data source and a degree of interest for each tuple of interest;

sampling tuples from said tuples of interest into said representation based upon said degree of interest in a manner that preserves at least a subset of said relationships among tuples in the data source; and

storing values in the representation that relate to the likelihood that each tuple sampled into said representation would be sampled into the representation if the sampling process were to be repeated.

79. A system as claimed in claim 78 wherein said software implements a designer component for:

interacting with a user; and

defining parameters used to construct said representation based upon said parameters.

80. A system as claimed in claim 79 wherein:

said designer component provides a user with a list of distinct valid values of categorical attributes from dimension defining tables and/or a list of valid value ranges for real-valued attributes; and

those subsets of tuples in said data source not associated with categorical values or value ranges that are selected by the user are marked for exclusion from said representation.

81. A system as claimed in claim 78 wherein said software implements a designer component for:

interacting with a user; and

defining parameters used to construct a collection of scaled representations based upon said parameters.

82. A system as claimed in claim 81 wherein said software is configured to construct a collection of scaled representations by first constructing a largest representation and then subsampling said largest representation.

83. A system as claimed in claim 78 wherein said software implements a designer component for interacting with a user to allow said user to adjust the balance of tuples in said representation and said software constructs said representation based upon said adjustment.

84. A system as claimed in claim 78 wherein said software implements an analyst component for:

intercepting an original query;

remapping said original query into a format compatible with said representation;

applying said remapped query against said representation; and

providing the results of the remapped query in response to said original query.

85. A system as claimed in claim 84 wherein said results of the remapped query include one or more approximate answers.

86. A system as claimed in claim 85 wherein said results of the remapped query include a variance with each approximate answer.

87. A system as claimed in claim 85 wherein said results of the remapped query include a confidence interval for the exact answer with each approximate answer.

88. A system as claimed in claim 84 wherein said software implements a builder component for constructing multiple representations of said data source and said analyst component is further configured for selecting between said multiple representations to select an optimal representation from said multiple representations to apply said remapped query against.

89. A system as claimed in claim 88 wherein said software is further configured to construct multiple scaled versions of said representation and said software is further capable of applying said remapped query against a select one of said multiple scaled versions of said representation.

90. A system as claimed in claim 88 wherein said multiple representations constructable by said builder component are selected from the group consisting of sampling, pre-computed aggregates, histograms, wavelets, data cubes, and data clouds.

91. A system as claimed in claim 78 wherein said software implements a reporter component for outputting one or more approximate answers to said original query.

92. A system as claimed in claim 91 wherein said reporter component optionally outputs a variance with each approximate answer.

93. A system as claimed in claim 91 wherein said variance is provided by the reporter component as hidden metadata.

94. A system as claimed in claim 91 wherein said reporter component optionally outputs a confidence interval for the exact answer with each approximate answer.

95. A system as claimed in claim 94 wherein said confidence interval is provided by the reporter component as hidden metadata.

96. A computer readable medium including program code representing computer implemented operations for constructing a representation from a data source in order to provide relatively quick response to queries related to information in said data source, wherein said data source has a plurality of tuples stored in said data source and a data

source schema that includes defined relationships among at least a subset of the tuples in the data source, said operations comprising:

creating said representation by copying at least a subset of said data source schema to define a representation schema;

adding additional data to said representation that represents information that is not in said data source;

defining tuples of interest within said data source and a degree of interest for each tuple of interest;

sampling tuples from said tuples of interest into said representation based upon said degree of interest in a manner that preserves at least a subset of said relationships among tuples in the data source; and

storing values in the representation that relate to the likelihood that each tuple sampled into said representation would be sampled into the representation if the sampling process were to be repeated.

97. A method for translating simple SQL queries directed at sampling initiation and ancestor-sampled tables of a data source into revised SQL queries directed at an Information Reservoir or Information Reservoir collection created from said data source in order to calculate both approximate query answers and variances for the approximate answers, said method comprising:

comparing said simple SQL query to a list containing both SQL query types that can be translated and the associated translation rule or rules to be applied for each SQL query type that can be translated; and

applying said translation rule or rules associated with said simple SQL query to translate said simple SQL query into a revised query directed at said Information Reservoir or Information Reservoir collection created from said data source.

98. The method as claimed in claim 97 wherein said translation rules are text substitution rules.

99. The method as claimed in claim 97 wherein said simple SQL query can include an aggregate expression composed of linear combinations of simple aggregate functions directed at directly-sampled tables.

100. The method as claimed in claim 97 wherein said method is computer-implemented and said comparing and said translating are performed automatically.

101. A computer-implemented method for translating queries directed at a data source into revised queries directed at an Information Reservoir or Information Reservoir collection created from said data source in order to calculate both approximate query answers and variances for the approximate answers, said method comprising:

translating queries directed at said data source into a sequence of atomic operations that act on said data source; and

translating atomic operations that act on said data source to atomic operations that act on said Information Reservoir or Information Reservoir collection in order to calculate both approximate query answers and variances for the approximate answers.

102. A computer-implemented method as claimed in claim 101 further comprising the optional translation of atomic operations on said Information Reservoir or Information Reservoir collection to queries on said Information Reservoir or Information Reservoir collection.

103. A computer-implemented method as claimed in claim 101 further comprising a structure for storing table metadata for each table in said Information Reservoir or Information Reservoir collection, said table metadata comprising table names and aliases, foreign and primary keys, lists of attributes along with attribute sampling type, attribute variance, and location of associated rate of inclusion.

104. A computer-implemented method as claimed in claim 103 wherein said table metadata is defined not only for tables in said Information Reservoir or Information

Reservoir collection, but also for each table that results from a query or atomic query operation applied to one or more tables of said Information Reservoir or Information Reservoir collection.

105. The computer-implemented method as claimed in claim 103 wherein said structure contains the schema of said Information Reservoir or Information Reservoir collection, said schema being optionally augmented with each table that is the result of a query or atomic query operation applied to one or more tables of said Information Reservoir or Information Reservoir collection.

106. The computer-implemented method as claimed in claim 105, wherein said queries directed at said data source may be translated to queries or atomic query operations directed at tables that resulted from previous queries or atomic query operations applied to one or more tables of said Information Reservoir or Information Reservoir collection.

107. The computer-implemented method as claimed in claim 101 further comprising a structure containing data necessary for determining which translation to apply during the query translation process.

108. The computer-implemented method as claimed in claim 107 where said data is a rule set comprising:

formulas for computing approximate query answers and variances for the approximate answers;

translation rules for replacing an atomic operation with a sequence of atomic operations; and

rules for updating table metadata and augmenting table collection schema.

109. A query system for use with an Information Reservoir or Information Reservoir collection created from a data source comprising at least one processor programmed to:

translate queries directed against said data source into a sequence of atomic operations that act on said data source; and

translate atomic operations that act on said data source to atomic operations that act on said Information Reservoir or Information Reservoir collection in order to calculate both approximate query answers and variances for the approximate answers.

110. A query system as claimed in claim 109 wherein said at least one processor is further programmed to optionally translate atomic operations on said Information Reservoir or Information Reservoir collection to queries on said Information Reservoir or Information Reservoir collection.

111. A computer readable medium including program code representing computer implemented operations for constructing a representation from a data source in order to provide relatively quick response to queries related to information in said data source, wherein said data source has a plurality of tuples stored in said data source and a data source schema that includes defined relationships among at least a subset of the tuples in the data source, said operations comprising:

creating said representation by copying at least a subset of said data source schema to define a representation schema;

adding additional data to said representation that represents information that is not in said data source;

defining tuples of interest within said data source and a degree of interest for each tuple of interest;

sampling tuples from said tuples of interest into said representation based upon said degree of interest in a manner that preserves at least a subset of said relationships among tuples in the data source; and

storing values in the representation that relate to the likelihood that each tuple sampled into said representation would be sampled into the representation if the sampling process were to be repeated.